UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/040,500	12/28/2001	David J. Long	50277-1767	3700

			EXAMINER
29989 7590 01/17/2006			DAO, THUY CHAN

HICKMAN PALERMO TRUONG & BECKER, LLP
2055 GATEWAY PLACE
SUITE 550
SAN JOSE, CA 95110

ART UNIT	PAPER NUMBER
2192	

DATE MAILED: 01/17/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

	Application No.	Applicant(s)
Office Action Summary	10/040,500	LONG ET AL.
	Examiner	Art Unit	
	Thuy Dao	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _03_ MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _03 October 2005_.

2a)☒ This action is **FINAL**.   2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☐ Claim(s) _____ is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-45_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on _28 December 2001_ is/are: a)☒ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All   b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _10/31 & 11/21/05_.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____ .

U.S. Patent and Trademark Office
PTOL-326 (Rev. 1-04)   Office Action Summary   Part of Paper No./Mail Date 3

DETAILED ACTION

1. This action is responsive to the Reply after Final Action filed on October 3, 2005.

2. Claims 1-45 are presented and independent claims 1 and 24 have been amended.

Information Disclosure Statement

3. The information disclosure statement (IDS) submitted on October 31, and November 21 were filed after the mailing date of the Final Action on August 25, 2005. The submission is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statements are being considered by the examiner.

Priority Date

4. For the record, the Applicant claimed continuation priority date of US Patent Application Serial No. 09/853,823 filed May 11, 2001 (Specification, page 1, paragraph 1), which claims domestic priority from US Provisional Application Serial No. 60/204,196 filed May 12, 2000.

Disclosure Must Be the Same: The later-filed application must be an application for a patent for an invention which is also disclosed in the prior application (the parent or original nonprovisional application or provisional application); the disclosure of the invention in the parent application and in the later-filed application must be sufficient to comply with the requirements of the first paragraph of 35 U.S.C. 112. See Transco Products, Inc. v. Performance Contracting, Inc., 38 F.3d 551, 32 USPQ2d 1077 (Fed. Cir. 1994).

As set forth in the previous Office Action, the continuation priority date was not acknowledged because the disclosures of the instant application and the application 09/853,823 are not the same.

5. The priority date of the US Provisional Application Serial No. 60/204,196 (May 12, 2000) is not acknowledged because the instant application's earliest effective filling date was December 28, 2001 and later than 12 months after the date on which the

provisional application was filed. See MPEP, Appendix L, section 35 USC 119(e), page 24.

6. The priority date of this application is the filing date December 28, 2001.

Response to Amendments

7. The Applicant amended the claims and included "... *said data item is an instance of a first class and inherits attributes and methods from said first class*" into independent claims 1 and 24.

8. Since this application is eligible for the transitional procedure of 37 CFR 1.129(a), and the fee set forth in 37 CFR 1.17(r) has been timely paid, the finality of the previous Office action is hereby withdrawn pursuant to 37 CFR 1.129(a). Applicant's submission after final filed on October 3, 2005 has been entered.

Claim Objections

9. Claim 24 is objected to because of the following informality in line 5, which limitation the Examiner presumes and/or treats as --... wherein said first category class has one or more attributes;-- (see claim 1, lines 5-6).

        Appropriate correction is required.

Claim Rejections – 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

11. Claims 1-13, 16-36, and 39-45 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chau (art of record, US Patent No. 6,721,727) in view of US Publication No. 2002/0073091 A1 to Jain et al. (art made of record, hereinafter "Jain") and further in view of US Patent No. 6,567,819 to Cheng et al. (art made of record, hereinafter "Cheng").

Claim 1:

Chau discloses a computer-implemented method for establishing a structure of a data item (order.xml) within a computer system, the method comprising the steps of:

creating a category object (e.g., columns 11-13, LineItem.dtd/dad.dtd), *wherein said category object has one or more attributes* and

associating said data item (order.xml) *with said category object without associating said category object with all other instances* (e.g., part.xml, sales.xml, price.xml) *thereby causing said data item to be associated with a structure that includes storage for values for said one or more attributes of said category object* (e.g., column 7, lines 26-51, "XML System provides good data and metadata management solutions to handle traditional and non-traditional data. With the content of structured XML documents in a database, a user can combine structured XML information with traditional relational data. Based on the application, a user can choose whether to store entire XML documents in a database as a non-traditional distinct data type or map the XML content as traditional data in relational tables. For non-traditional XML data types, the XML System adds the power to search rich data types of XML element or attribute values. For traditional SQL data, that is either decomposed from incoming XML documents or in existing relational tables to be used to create outgoing XML documents, the XML System provides a custom mapping mechanism to allow the transformation between XML documents and relational data.

The XML System offers the flexibility to store entire XML documents as column data or transform between XML documents and data in existing tables. The transformation includes decomposing an XML document into one or multiple pieces and storing the pieces in the form of relational data, as well as, composing XML documents from the data in existing relational tables. A user can decide how structured XML documents are to be stored or created through a Document Access Definition (DAD)").

Chau does not explicitly disclose said XML file *is an instance of a first class and inherits attributes and methods from said first class* and *said category object is an instance of a category class.*

However, in an analogous art of translating XML document to an object in an object-oriented language, Jain discloses said XML file is an instance of a first class and inherits attributes from said first class and said category object is an instance of a category class (e.g., page 2, paragraph [028], "As mentioned above, a translation tool is provided for converting <u>an XML DTD associated with the XML document to Java classes</u> from which a Java object corresponding to the XML document is instantiated. In accordance with one embodiment, the translation tool first uses a parser to read the structure of the XML DTD and thus identifies all nodes in the XML DTD. <u>For each of the identified nodes in the XML DTD, the translation tool creates a corresponding Java class and writes this Java class to a respective file</u>. For each of the created Java classes, the translation tool initially generates a package statement. The translation tool then generates the standard import statements"; and

page 2, paragraph [030], "FIG. 1 shows elements that are used in a technique for <u>converting an XML document 102 to a Java object 124</u>, in accordance with one embodiment of the invention. In the XML domain, each XML document 102 is associated with an XML DTD 104. XML_to_Java translation tool 108 is used to convert an XML DTD 104 to Java classes 110 and stores each of these classes 110 in a respective file 112. Those skilled in the art will recognize that one Java class 110 is stored in one file 112 to practice modular programming. However, depending on the implementation, all Java classes 110 may be stored in one file 112 or various files 112 without departing from the scope of the various embodiments of the invention. The invention is not limited to how the classes 110 are stored in files 112. Each of the Java files 112 contains a Java class that is written in the Java language" (emphasis added)).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine the teachings of Jain into Chau's. One would have been motivated to do so because that would provide an enhanced system for interfacing the XML document more flexible with such a well-known Java-based environment and work with the content of the XML document more efficiently as once suggested by Jain (e.g., page 1, paragraphs [011-013]).

Chau and Jain do not explicitly disclose methods in said first class. However, in an analogous art of creating object-oriented class from XML files, Cheng discloses technique to make an instance of a class, defined by using XML, also has methods (e.g., column 4, Table I, class "ebObj.Purchase"; column 6, Table II, object with Purchase_ID AXN10009; and column 8, Table III, object with Purchase_ID 0-5001-1001 and execution of method closeDeal).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine the teachings of Cheng into Chau's and Jain's. One would have been motivated to do so to answer the need to make rapid changes of objects and classes when mapping information to a database and enhance the conventional database software (e.g., RDBMS) as once disclosed by Cheng (e.g., column 2, lines 9-31; column 3, lines 25-38).

Claim 2:

The rejection of base claim 1 is incorporated. Chau also discloses *the computer-implemented steps of:*

receiving data that is designated for a particular attribute of said one or more attributes;

determining whether said data conforms to rules associated with said particular attribute; and

if said data conforms to said rules

storing said data as a value into said particular attribute.

These limitations have been addressed and/or set forth in the followings (e.g., FIG. 2 and related text, column 5, line 66 to column 6, line 23, "FIG. 2 is a diagram illustrating a computer hardware environment that could be used in accordance with the present invention. In one embodiment, the DB2 XML Extender 200, a product from International Business Machines, Corporation, is at the center of the architecture. An application program 202 and a document access definition (DAD) 204 are received by the DB2 XML Extender 200. The DB2 XML Extender 200 takes an XML document 206 as the input, stores the XML document 206 in DB2 210 (i.e., a relational database)

either internally inside DB2 210 or externally on the file system as one or more XML files 208. Then, the stored XML document 206 can also be retrieved from DB2 210 or the file system through the DB2 Extender 200. The processing performed by the DB2 XML Extender 200 will be described in more detail below.

In another embodiment, an application program 202 and a document access definition (DAD) 204 are received by the DB2 XML Extender 200. The DB2 XML Extender 200 takes an XML document 206 as input, decomposes the XML document 206 into fragmented data and stores the fragmented data in DB2 210 (i.e., a relational database). Then, the fragmented data stored in DB2 210 can be regenerated from DB2 210 through the DB2 Extender 200. The processing performed by the DB2 XML Extender 200 will be described in more detail below").

Claim 3:

The rejection of base claim 2 is incorporated. Chau also discloses *the step of receiving, determining, and storing are performed by a method in said category class* (e.g., column 6, lines 6-10, "The DB2 XML Extender 200 takes an XML document 206 as the input, stores the XML document 206 in DB2 210 (i.e., a relational database) either internally inside DB2 210 or externally on the file system as one or more XML files 208"; and

FIG. 11 and related text, column 75, lines 37-49).

Claim 4:

The rejection of base claim 2 is incorporated. Chau also discloses *said rules are data type rules associated with a data type of said particular attribute* (e.g., column 7, lines 45-51, "...The transformation includes decomposing an XML document into one or multiple pieces and storing the pieces in the form of relational data, as well as, composing XML documents from the data in existing relational tables. A user can decide how structured XML documents are to be stored or created through a Document Access Definition (DAD)").

Claim 5:

The rejection of base claim 2 is incorporated. Chau also discloses *said rules are software rules* (e.g., column 9, lines 1-8, "The XML System also serves as an XML document type definition (DTD) repository. When a database is XML enabled, a DTD Reference Table (DTD_REF) is created. Each row of this table represents a DTD, with additional metadata information. This table is accessible by users, and allows them to insert their own DTDs. The DTDs in the DTD_REF table are used to validate XML documents and to help applications to define a document access definition (DAD)").

Claim 6:

The rejection of base claim 1 is incorporated. Chau also discloses *the step of:*

storing within a database, objects that define said data item and said category object (e.g., column 7, lines 26-51, as set forth in claim 1 above).

Claim 7:

The rejection of base claim 1 is incorporated. Chau also discloses *the computer-implemented step of:*

maintaining an object relational mapping system that indicates a correlation between said data item and data stored in a relational database (e.g., column 75, lines 37-49, "FIG. 11 is a flow diagram illustrating the steps performed by the XML System to decompose XML documents with application specific mappings. In block 1100, the XML System receives an XML document containing XML data. In block 1102, the XML System parses the XML document to generate an XML Document Object Model (DOM) tree. In block 1104, the XML System receives a data access definition (DAD) that identifies one or more relational tables and columns. In block 1106, the XML System processes the DAD to generate a DAD Document Object Model (DOM) tree. In block 1108, the XML System maps data from the XML DOM tree to columns in relational tables according to the DAD DOM tree").

Claim 8:

The rejection of base claim 1 is incorporated. Chau also discloses *said category class is a user defined subclass of a parent category class* (e.g., column 9, lines 1-8, "The XML System also serves as an XML document type definition (DTD) repository. When a database is XML enabled, a DTD Reference Table (DTD_REF) is created. Each row of this table represents a DTD, with additional metadata information. This table is accessible by users, and allows them to insert their own DTDs. The DTDs in the DTD_REF table are used to validate XML documents and to help applications to define a document access definition (DAD)").

Claim 9:

The rejection of base claim 1 is incorporated. Chau also discloses *the step of associating said data item with said category object further includes the computer-implemented step of:*

> *establishing a pointer from said category object to said data item* (e.g., column 3, lines 16-21, "... Initially, an XML document containing XML data is received. A document access definition that identifies one or more relational tables and columns is received. The XML data is mapped from the application DTD to the relational tables and columns using the document access definition based on the XPath data model").

Claim 10:

The rejection of base claim 1 is incorporated. Chau also discloses *the step of associating said data item with said category object further includes the computer-implemented step of:*

> *maintaining a table that includes an entry that indicates that said data item is associated with said category* (e.g., FIG. 4 and related text, column 34, lines 19-37, "FIG. 4 is a flow diagram illustrating steps performed by the XML System in creating and maintaining XML document data as column data. In Block 400, the XML System creates a table with an XML column having a XML column type. The table is created in response to a CREATE TABLE statement that specifies the XML column. In Block 402, the XML System enables the XML column. Next, the XML System, in Block 404, creates

side tables using a Data Access Definition for the XML column. In Block 406, the XML System creates triggers for Insert, Update, and Delete on the XML column, so that the side tables are populated when the main table is populated and the side tables are modified when the main table is modified. Thus, the main table and side tables are synchronized. In Block 408, when data is inserted into the main table, the XML System inserts data into the side tables. In Block 410, when the main table is modified (i.e., data is updated or deleted), the XML System modifies the side tables").

Claim 11:

The rejection of base claim 10 is incorporated. Chau also discloses *the step of maintaining a table further includes the computer-implemented step of:*

maintaining said entry to include a key that identifies said category object and a pointer to said category object (e.g., FIG. 3 and related text, column 18, line 66 to column 19, line 3, "FIG. 3 illustrates an application or main table and its four side tables. The Application table 300 has a root_id in common with each side table 302, 304, 306, and 308. The side tables 302, 304, 306, and 308 correspond to the side tables defined in the DAD above").

Claim 12:

The rejection of base claim 10 is incorporated. Chau also discloses *the step of maintaining a table further includes the computer-implemented step of:*

maintaining said table externally to said data item (e.g., FIG. 3 and related text, column 18, line 66 to column 19, line 3, "FIG. 3 illustrates an application or main table and its four side tables. The Application table 300 has a root_id in common with each side table 302, 304, 306, and 308. The side tables 302, 304, 306, and 308 correspond to the side tables defined in the DAD above").

Claim 13:

The rejection of base claim 10 is incorporated. Chau also discloses *the step of maintaining a table further includes the computer-implemented step of:*

maintaining said table internally to said data item (e.g., column 37, lines 1-22,

order key	customer name	customer email	part key	color	qty	price	tax	ship_id	date	mode
1	General Motor	parts@ gm.com	68	red	36	34850.16	0.06	4.58.825484	1998-08-19	BOAT
1	General Motor	parts@ gm.com	68	red	36	34850.16	0.06	4.58.825537	1998-08-19	AIR
1	General Motor	parts@ gm.com	128	red	28	34000.00	0.07	4.58.825589	1998-12-30	TRUCK

The data in order_key, customer_name, customer_email, part_key, qty, price, and tax are duplicated for each shipment. The data in order_key, customer_name, and customer_email are duplicated for each part. This issue is addressed by partitioning the columns into equivalence classes that reflect the semantics of the relational data: [order_key, customer_name, customer_email], [part_key, color, qty, price, tax], and [ship_id, date, mode]. The XML System opens a new cursor only when it crosses a boundary between classes").

Claim 16:

The rejection of base claim 1 is incorporated. Chau also discloses *the step of creating a category object further includes the computer-implemented step of:*

maintaining a table that includes an entry that contains a particular attribute of said one or more attributes (e.g., FIG. 8-9 and related text, column 49, lines 8-42, "FIG. 8 is a block diagram illustrating components of the XML System in one embodiment of the invention. Relational tables 800 store relational data. A Document Access Definition (DAD) 802 defines an Xcollection 804 and a SQL query 806. A Document Type Definition (DTD) 808 is used to validate and define the DAD 802. The

SQL query is used to retrieve data from the relational tables 800. Using the DAD 802, the SQL query 806, and the XML composition stored procedures 810, the XML system generates one or more XML documents 812. The XML system stores the data used to generate the one or more XML documents in an XML Collection table 814. Although the relational tables and XML Collection tables are shown in different data storage devices 800 and 814, both types of tables could reside at one data storage device.

FIG. 9 ... In particular, the XML System uses these components to map each column of the retrieved data to an XML element or attribute. Then, the XML System stores the data used to generate the one or more XML documents in an XML Collection. One skilled in the art would recognize that the one or more XML documents could be stored in another manner, for example, in other types of tables or as a file").

Claim 17:

The rejection of base claim 16 is incorporated. Chau also discloses *the step of maintaining a table further includes the computer-implemented step of:*

maintaining said entry to include a key that identifies said particular attribute.

Claim 17 recites the same limitations as those of the method claim 11, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the reference teaches all of the limitations of claim 11, it also teaches all of the limitations of claim 17.

Claim 18:

The rejection of base claim 16 is incorporated. Chau also discloses *the step of maintaining a table further includes the computer-implemented step of:*

maintaining said table externally to said category object.

Claim 18 recites the same limitations as those of the method claim 12, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the reference teaches all of the limitations of claim 12, it also teaches all of the limitations of claim 18.

Claim 19:

The rejection of base claim 16 is incorporated. Chau also discloses *the step of maintaining a table further includes the computer-implemented step of:*

maintaining said table internally to said category object.

Claim 19 recites the same limitations as those of the method claim 13, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the reference teaches all of the limitations of claim 13, it also teaches all of the limitations of claim 19.

Claim 22:

The rejection of base claim 1 is incorporated. Chau also discloses *the computer-implemented step of:*

associating said category object with a second data item that is an instance of a second class, without associating said category object with all other instances of said second class wherein said first class is a different class from said second class (e.g., FIG. 3 and related text, column 18, line 66 to column 19, line 3, "FIG. 3 illustrates an application or main table and its four side tables. The Application table 300 has a root_id in common with each side table 302, 304, 306, and 308. The side tables 302, 304, 306, and 308 correspond to the side tables defined in the DAD above";

and column 37, lines 1-22,

order key	customer name	customer email	part key	color	qty	price	tax	ship_id	date	mode
1	General Motor	parts@ gm.com	68	red	36	34850.16	0.06	4.58.825484	1998-08-19	BOAT
1	General Motor	parts@ gm.com	68	red	36	34850.16	0.06	4.58.825537	1998-08-19	AIR

1	General Motor	parts@ gm.com	128	red	28	34000.00	0.07	4.58.825589	1998-12-30	TRUCK

"The data in order_key, customer_name, customer_email, part_key, qty, price, and tax are duplicated for each shipment. The data in order_key, customer_name, and customer_email are duplicated for each part. This issue is addressed by partitioning the columns into equivalence classes that reflect the semantics of the relational data: [order_key, customer_name, customer_email], [part_key, color, qty, price, tax], and [ship_id, date, mode]. The XML System opens a new cursor only when it crosses a boundary between classes").

Claim 23:

The rejection of base claim 1 is incorporated. Chau discloses the framework with XML, DTD, and database files (e.g., column 7, lines 26-51; columns 11-13) but does not explicitly disclose classes in the system.

However, Jain discloses *said category class is a first file type* (DTD class file type) *and said category object is a first file* (e.g., columns 11-13, LineItem.dtd/dad.dtd) *of said first file type in a file system*;

wherein said first class is a second file type (XML class file type) *and said data item is a second file (order.xml) of said second file type in a file system*; and

wherein the step of associating includes associating said second file (order.xml) *with said first file* (LineItem.dtd/dad.dtd) *without associating said first file with all other instances* (e.g., part.xml, sales.xml, price.xml) *of said second file type* (XML class file type) *thereby causing said second file* (order.xml) *to be associated with said structure in said file system* (e.g., page 2, paragraphs [228-030]).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine the teachings of Chau and Jain. One would have been motivated as set forth in claim 1 above.

Claim 24:

Chau discloses *a method for establishing a structure of a data item* (e.g., column 12, file order.xml) *within a computer system, the method comprising the steps of:*

creating a first category object (e.g., columns 11-13, LineItem.dtd/dad.dtd) *herein said first category object has one or more attributes;*

creating a second category object (e.g., a tuple in table "order") *that is an instance of a second category class* (table "order"),

associating said data item (order.xml) *with said first category object* (e.g., dad.dtd) *and with said second category object* (tuple in table "order") *thereby causing said data item to be associated with a structure that includes storage for values for said one or more attributes of said first category object and for said one or more attributes of said second category object* (e.g., FIG. 2 and related text, column 5, line 66 to column 6, line 23, "FIG. 2 is a diagram illustrating a computer hardware environment that could be used in accordance with the present invention. In one embodiment, the DB2 XML Extender 200, a product from International Business Machines, Corporation, is at the center of the architecture. An application program 202 and a document access definition (DAD) 204 are received by the DB2 XML Extender 200. The DB2 XML Extender 200 takes an XML document 206 as the input, stores the XML document 206 in DB2 210 (i.e., a relational database) either internally inside DB2 210 or externally on the file system as one or more XML files 208. Then, the stored XML document 206 can also be retrieved from DB2 210 or the file system through the DB2 Extender 200. The processing performed by the DB2 XML Extender 200 will be described in more detail below.

In another embodiment, an application program 202 and a document access definition (DAD) 204 are received by the DB2 XML Extender 200. The DB2 XML Extender 200 takes an XML document 206 as input, decomposes the XML document 206 into fragmented data and stores the fragmented data in DB2 210 (i.e., a relational database). Then, the fragmented data stored in DB2 210 can be regenerated from DB2 210 through the DB2 Extender 200. The processing performed by the DB2 XML Extender 200 will be described in more detail below").

Chau does not explicitly disclose said XML file is *an instance of a first class and inherits attributes and methods form said first class* and *said category object is an instance of a category class.*

However, in an analogous art of translating XML document to an object in an object-oriented language, Jain discloses said XML file is an instance of a first class and inherits attributes form said first class and said category object is an instance of a category class (e.g., page 2, paragraph [028], "As mentioned above, a translation tool is provided for converting <u>an XML DTD associated with the XML document to Java classes</u> from which a Java object corresponding to the XML document is instantiated. In accordance with one embodiment, the translation tool first uses a parser to read the structure of the XML DTD and thus identifies all nodes in the XML DTD. <u>For each of the identified nodes in the XML DTD, the translation tool creates a corresponding Java class and writes this Java class to a respective file</u>. For each of the created Java classes, the translation tool initially generates a package statement. The translation tool then generates the standard import statements"; and

page 2, paragraph [030], "FIG. 1 shows elements that are used in a technique for <u>converting an XML document 102 to a Java object 124</u>, in accordance with one embodiment of the invention. In the XML domain, each XML document 102 is associated with an XML DTD 104. XML_to_Java translation tool 108 is used to convert an XML DTD 104 to Java classes 110 and stores each of these classes 110 in a respective file 112. Those skilled in the art will recognize that one Java class 110 is stored in one file 112 to practice modular programming. However, depending on the implementation, all Java classes 110 may be stored in one file 112 or various files 112 without departing from the scope of the various embodiments of the invention. The invention is not limited to how the classes 110 are stored in files 112. Each of the Java files 112 contains a Java class that is written in the Java language (emphasis added)").

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine the teachings of Jain into Chau's. One would have been motivated to do so because that would provide an enhanced system for interfacing the XML document more flexible with such a well-known Java-based environment and

work with the content of the XML document more efficiently as once suggested by Jain (e.g., page 1, paragraphs [011-013]).

Chau and Jain do not explicitly disclose methods in said first class. However, in an analogous art of creating object-oriented class from XML files, Cheng discloses technique to make an instance of a class, defined by using XML, also has methods (e.g., column 4, Table I, class "ebObj.Purchase"; column 6, Table II, object with Purchase_ID AXN10009; and column 8, Table III, object with Purchase_ID 0-5001-1001 and execution of method closeDeal).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine the teachings of Cheng into Chau's and Jain's. One would have been motivated to do so to answer the need to make rapid changes of objects and classes when mapping information to a database and enhance the conventional database software (e.g., RDBMS) as once disclosed by Cheng (e.g., column 2, lines 9-31; column 3, lines 25-38).

Claims 25-36 and 39-44:

The rejection of base claim 24 is incorporated. Claims 25-36 and 39-44 recite the same limitations as those of the claims 2-13 and 16-21, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the reference teaches all of the limitations of claims 2-13 and 16-21, it also teaches all of the limitations of claims 25-36 and 39-44.

Claim 45:

The rejection of base claim 24 is incorporated. Chau discloses the framework with XML, DTD, and database files (e.g., column 7, lines 26-51; columns 11-13) but does not explicitly disclose classes in the system.

However, Jain further discloses:

wherein said first category class is a first file type (DTD class file type) *and said first category object is a first file* (LineItem.dtd/dad.dtd) *of said first file type in a file system;*

wherein said second category class is a second file type (RDBMS file type) *and said second category object* (e.g., tuple in table "order") *is a second file of said second file type in said file system;*

wherein said class is a third file type (XML class file type) *and said data item is a third file* (order.xml) *of said third file type in said file system;* and

wherein the step of associating includes associating said third file with said first file and said second file thereby causing said third file to be associated with said structure in said file system that includes storage for values for said one or more attributes of said first file type and for said one or more attributes of said second file type (as addressed and set forth above).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine the teachings of Chau and Jain. One would have been motivated as set forth in claim 24 above.

12. Claims 14-15 and 37-38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chau in view of Jain and Cheng and further in view of Ng (art of record, US Patent No. 6,385,618) and Bowen (art of record, US Patent No 6,094,649).

Claim 14:

The rejection of base claim 1 is incorporated. Chau discloses *the step of associating said data item with said category object further includes the computer-implemented step of:*

locating an entry for said category object based on data associated with said category object; locating a pointer to said category object in said entry and following said pointer to locate said category object (e.g., column 25, lines 24-39, "Since the DAD is specified by the application, the side tables created by the XML System are known to the application programmer. For better performance, an application can do query or sub-query on side tables directly. The following example shows how to do so for the same query stated above:

SELECT sales_person from sales_tab

> WHERE invoice_number in
>
> (SELECT invoice_number from part_tab
>
> WHERE price>2500.00)

Note that the invoice_number is the primary key in the application table sales_tab. The advantage of direct query with sub-query is better performance. When side tables have parent-children relationships, direct query with sub-query often make more sense").

Chau does not explicitly discloses

> *storing into said data item a hash table; and*
>
> *locating an entry in said hash table.*

However, in an analogous art of database mapping tool, Ng. discloses storing into said data item a hash table (e.g., FIG. 3, FIG. 4A and related text, column 5, line 36 to column 6, line 32).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to modify the teachings of Chau by storing into said item a hash table as taught by Ng. One would have been motivated to index a lot of duplicate records as suggested by Ng. (e.g., column 5, lines 60-67).

Chau, Jain, Cheng, and Ng. do not explicitly disclose locating said entry in said hash table.

However, in an analogous art of searching tools for structured databases, Bowen discloses locating said entry in said hash table (e.g., column 11, lines 20-27, "The index 214 and the indexing agent 212 may use B-trees, hashing, and other familiar data structures and operations to create or modify or extend the index 214. If the documents 210 are in HTML format and the agent 212 is a web crawler that only indexes meta content tag values then comprehensive indexing places all (or substantially all) data values in the meta content tags so they will be indexed by the agent 212").

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine these teachings. One would have been motivated to enhance the system by associating keywords in the textual representation of each selected item's data with that item's location identifier as once suggested by Bowen (e.g., column 11, lines 16-19).

Claim 15:

The rejection of base claim 14 is incorporated. Chau also disclose *the computer-implemented steps of:*

receiving data that is designated as a key for locating said entry in said hash table;

determining whether said data conforms to rules associated with said key; and if said data conforms to said rules

using said data as said key to locate said entry.

(e.g., column 25, lines 30-39,

"SELECT sales_person from sales_tab

WHERE invoice_number in

(SELECT invoice_number from part_tab

WHERE price>2500.00)

Note that the invoice_number is the primary key in the application table sales_tab. The advantage of direct query with sub-query is better performance. When side tables have parent-children relationships, direct query with sub-query often make more sense").

Claims 37 and 38:

The rejection of base claim 24 is incorporated. Claims 37 and 38 recite the same limitations as those of the claims 14 and 15, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the references teach all of the limitations of claims 14 and 15, they also teach all of the limitations of claims 37 and 38.

Conclusion

13. Applicant's amendment filed on June 10, 2005 necessitated the new ground(s) of rejection presented in this Office action. Accordingly, THIS ACTION IS MADE FINAL. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

14. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure:

"Virtual Methods Tables in Python", Martin von Lowis, discloses a set of class attributes (methods and data), which is shared among all instances, and the attribute can be a method or a data attribute, per-instance or defined in the class. Online version can be retrieved at <http://www.foretec.com/python/workshops/1998-11/proceedings /papers /lowis/lowis.html>.

US Patent Nos. 6,901,588, 6,594,823, and US Patent Publication Nos. 2002/0091702 A1, 2002/0035645 A1, and 2001/0047385 A1 disclose methods and systems for representing high-level programming language and XML data structures.

15. Any inquiry concerning this communication should be directed to examiner Thuy Dao (Twee), whose telephone is (571) 272 8570. The examiner can normally be reached on Monday – Friday from 6:30AM to 3:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam, can be reached at (571) 272 3695.
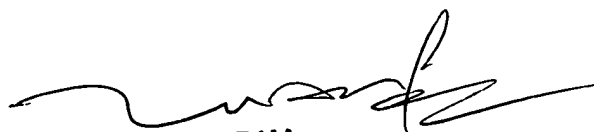
The fax phone number for the organization where this application or proceeding is assigned is (571) 273 8300.

Any inquiry of a general nature of relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is (571) 272 2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

T. Dao

TUAN DAM
SUPERVISORY PATENT EXAMINER